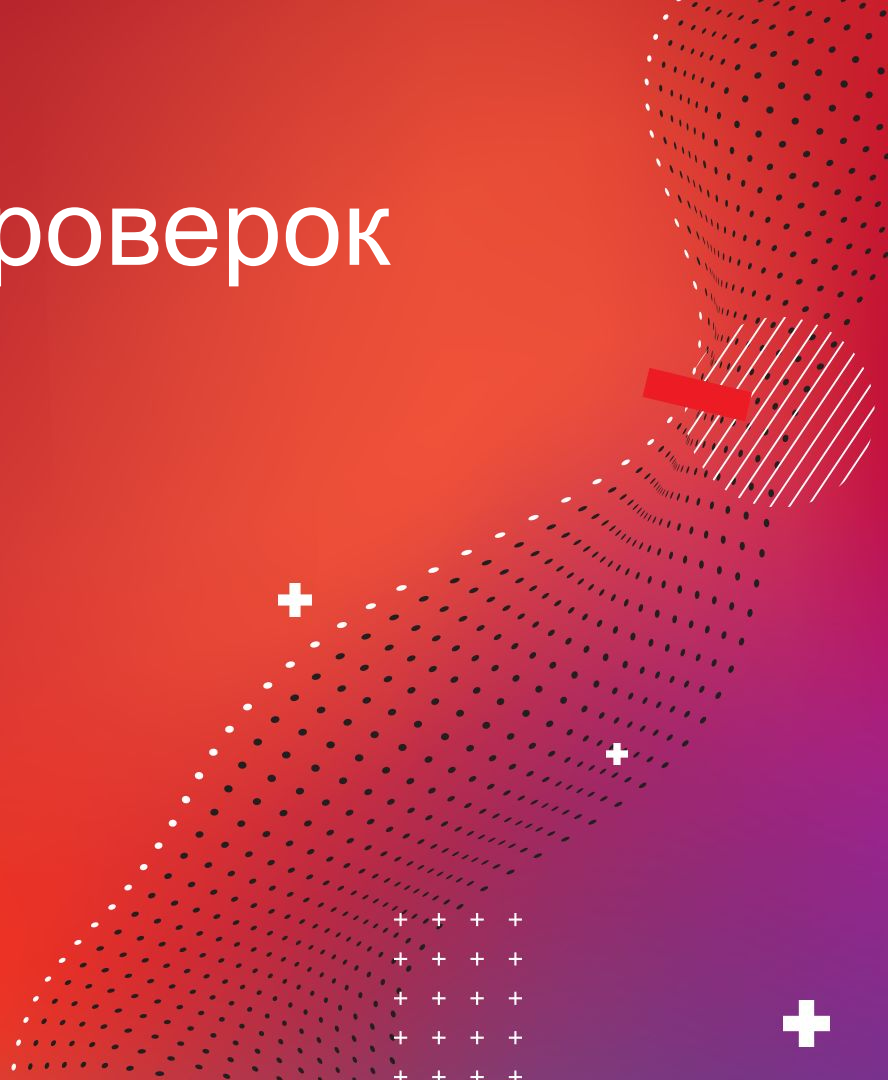


# Бурлаки деплоя. Автоматизация проверок финтех-систем

Васильев Федор



**HighLoad++**  
Весна 2021



# ДИСКЛЕЙМЕР

- © Внимание! Данная презентация настоятельно не рекомендуется к просмотру лицам не достигшим 18-летнего возраста, а также людям с неокрепшей психикой

# ДИСКЛЕЙМЕР

- ⊙ ~~Внимание! Данная презентация~~  
~~настоятельно не рекомендуется к~~  
~~просмотру лицам не достигшим 18-~~  
~~летнего возраста, а также людям с~~  
~~неокрепшей психикой~~
- ⊙ Поговорим о деплоях финтех-  
систем
- ⊙ Проиллюстрируем все это русской  
классической живописью



# ДИСКЛЕЙМЕР

- ⊙ ~~Внимание! Данная презентация~~  
~~настоятельно не рекомендуется к~~  
~~просмотру лицам не достигшим 18-~~  
~~летнего возраста, а также людям с~~  
~~неокрепшей психикой~~
- ⊙ Поговорим о деплоях финтех-  
систем
- ⊙ Проиллюстрируем все это русской  
классической живописью

В меру серьезно  
Достаточно просто  
Но с огоньком

# ДИСКЛЕЙМЕР

- ⊙ ~~Внимание! Данная презентация~~  
~~настоятельно не рекомендуется к~~  
~~просмотру лицам не достигшим 18-~~  
~~летнего возраста, а также людям с~~  
~~неокрепшей психикой~~
- ⊙ Поговорим о деплоях финтех-  
систем
- ⊙ Проиллюстрируем все это русской  
классической живописью
- ⊙ ... И на самом деле, мы уже  
начали!

В меру серьезно  
Достаточно просто  
Но с огоньком



\\ Чёрный супрематический квадрат \\  
Казимир Малевич

# Кто мы?

**350+**

IT-специалистов

**99,99%**

гарантированная  
надежность

**10 млн+**

транзакций в день

**5**

дата-центров по  
всему миру

 **ecommpay** — это про \$\$\$





\\ Последний день Помпеи \\  
Карл Брюллов

- ⊙ Эта история началась в самом конце 2019 года...
- ⊙ У нас был большой деплой перед новогодним фризом
- ⊙ Деплоили практически все компоненты
- ⊙ А некоторые несколько раз!





\\ Последний день Помпеи \\  
Карл Брюллов

- ⊙ Эта история началась в самом конце 2019 года...
- ⊙ У нас был большой деплой перед новогодним фризом
- ⊙ Деплоили практически все компоненты
- ⊙ А некоторые несколько раз!

И в чем, собственно, проблема?



\\ Последний день Помпеи \\  
Карл Брюллов

- © Мы работаем с финансовыми транзакциями, и потому у нас паранойя



\\ Последний день Помпеи \\  
Карл Брюллов

- © Мы работаем с финансовыми транзакциями, и потому у нас паранойя
- © Любой даунтайм или сбой в продакшне – это прямые фин. потери (помимо репутационных)



\\ Последний день Помпеи \\  
Карл Брюллов

- ⊙ Мы работаем с финансовыми транзакциями, и потому у нас паранойя
- ⊙ Любой даунтайм или сбой в продакшне – это прямые фин. потери (помимо репутационных)
- ⊙ Деплоим сервисы последовательно





\\ Последний день Помпеи \\  
Карл Брюллов

- ⊙ Мы работаем с финансовыми транзакциями, и потому у нас паранойя
- ⊙ Любой даунтайм или сбой в продакшне – это прямые фин. потери (помимо репутационных)
- ⊙ Деплоим сервисы последовательно
- ⊙ Каждый деплой тщательно проверяем



\\ Последний день Помпеи \\  
Карл Брюллов

- ⊙ Мы работаем с финансовыми транзакциями, и потому у нас паранойя
- ⊙ Любой даунтайм или сбой в продакшне – это прямые фин. потери (помимо репутационных)
- ⊙ Деплоим сервисы последовательно
- ⊙ Каждый деплой тщательно проверяем

Σ

Куча людей в течение 12 часов напряженно следят и деплоят...



\\ Последний день Помпеи \\  
Карл Брюллов

- ⊙ Мы работаем с финансовыми транзакциями, и потому у нас паранойя
- ⊙ Любой даунтайм или сбой в продакшне – это прямые фин. потери (помимо репутационных)
- ⊙ Деплоим сервисы последовательно
- ⊙ Каждый деплой тщательно проверяем

Σ

Куча людей в течение 12 часов напряженно следят и деплоят....

АВТОМАТИЗИРОВАТЬ!

ECOMMPAY.IT





\\ Последний день Помпеи \\  
Карл Брюллов

- ⊙ Мы работаем с финансовыми транзакциями, и потому у нас паранойя
- ⊙ Любой даунтайм или сбой в продакшне – это прямые фин. потери (помимо репутационных)
- ⊙ Деплоим сервисы последовательно
- ⊙ Каждый деплой тщательно проверяем

ЕЩЕ БОЛЬШЕ!

ECOMMPAY.IT

АВТОМАТИЗИРОВАТЬ!

ECOMMPAY.IT

Σ  
Куча людей в течение 12 часов  
напряженно следят и деплоят....



# Какой CI/CD нам нужен?

- ⊙ Чтобы работало
- ⊙ Быстро
- ⊙ Надежно
- ⊙ Дешево

# Какой CI/CD нам нужен?

- ⊙ Чтобы работало
  - ⊙ Быстро
  - ⊙ Надежно
  - ⊙ Дешево
- 
- ⊙ Чтобы работало
  - ⊙ Не было финансовых потерь во время и после деплоя
  - ⊙ А если и были, то маленькие
  - ⊙ И мы быстро их пресекли

# Какой CI/CD нам нужен?

- ⊙ Чтобы работало
  - ⊙ Быстро
  - ⊙ Надежно
  - ⊙ Дешево
- 
- ⊙ Чтобы работало
  - ⊙ Не было финансовых потерь во время и после деплоя
  - ⊙ А если и были, то маленькие
  - ⊙ И мы быстро их пресекли





\\ Супрематическая композиция \\

\\ Синий прямоугольник поверх красного луча \\

Казимир Малевич



\$ 85 млн !

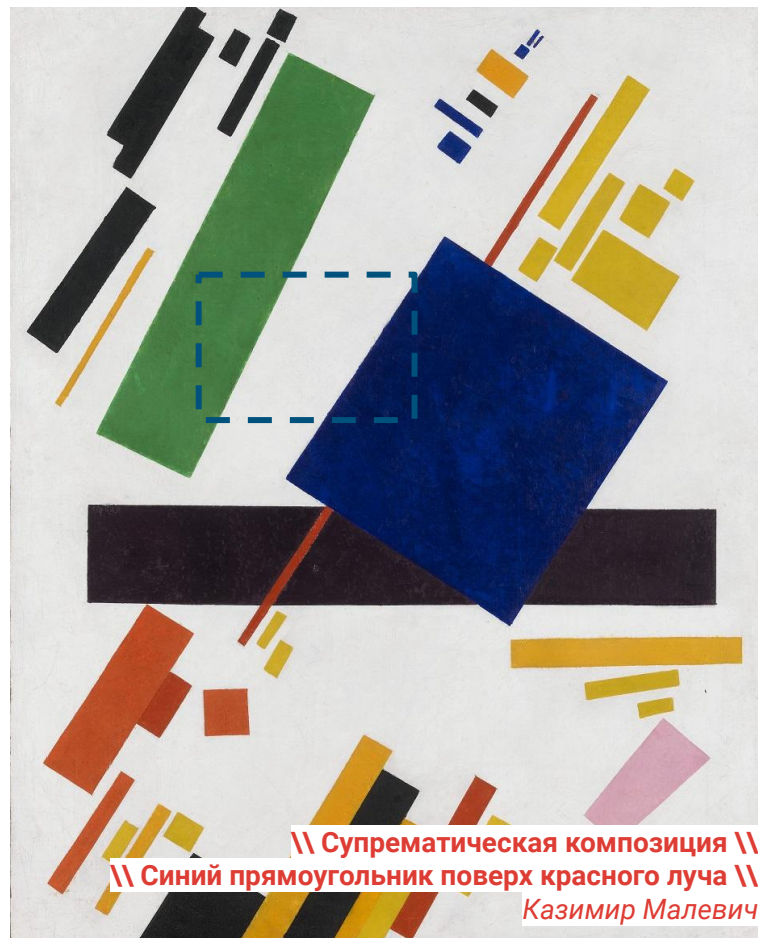


\\ Супрематическая композиция \\

\\ Синий прямоугольник поверх красного луча \\

Казимир Малевич

\$ 85 млн !



\\ Супрематическая композиция \\

\\ Синий прямоугольник поверх красного луча \\

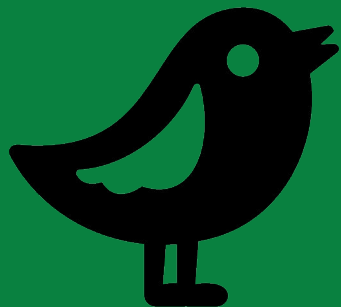
Казимир Малевич



# Blue \ Green Deployment

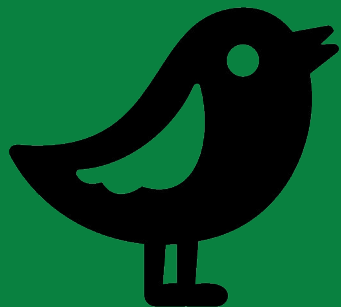
- ⊙ Для критичных сервисов вводим понятие “линии”
- ⊙ Катим новый код на офлайн-линию
- ⊙ Как-то проверяем
- ⊙ Если все нормально, переливаем трафик на линию с новым кодом





# Blue \ Green Deployment

- ⊙ Для критичных сервисов вводим понятие “линии”
- ⊙ Катим новый код на офлайн-линию
- ⊙ Как-то проверяем
- ⊙ Если все нормально, переливаем трафик на линию с новым кодом



# Blue \ Green Deployment

## Канареечные релизы

- ⦿ Трафик на офлайн-линию с новым кодом пускаем маленькими порциями: 2%, 5%, 20% ... 100%
- ⦿ Как-то проверяем на каждом шаге
  - ⦿ Для критичных сервисов вводим понятие “линии”
  - ⦿ Катим новый код на офлайн-линию
  - ⦿ Как-то проверяем
  - ⦿ Если все нормально, переливаем трафик на линию с новым кодом

# А как проверять?

- ⊙ Индивидуально (сервисы очень разные)

# А как проверять?

- ⊙ Индивидуально (сервисы очень разные)
- ⊙ Подробный письменный регламент проверки при деплое (обязательно)



# А как проверять?

- ⊙ Индивидуально (сервисы очень разные)
- ⊙ Подробный письменный регламент проверки при деплое (обязательно)
- ⊙ Смотрит человек (опытный инженер отдела мониторинга)

# А как проверять?

- ⦿ Индивидуально (сервисы очень разные)
- ⦿ Подробный письменный регламент проверки при деплое (обязательно)
- ⦿ Смотрит человек (опытный инженер отдела мониторинга)
- ⦿ Используем: Kibana, Grafana, Zabbix, руки и светлую голову 😊

# А как проверять?

- ⦿ Индивидуально (сервисы очень разные)
- ⦿ Подробный письменный регламент проверки при деплое (обязательно)
- ⦿ Смотрит человек (опытный инженер отдела мониторинга)
- ⦿ Используем: Kibana, Grafana, Zabbix, руки и светлую голову 😊
- ⦿ Помимо пунктов из регламента, рассматриваются аномалии в графиках/мониторах/дашбордах

# А как проверять?

- ⦿ Индивидуально (сервисы очень разные)
- ⦿ Подробный письменный регламент проверки при деплое (обязательно)
- ⦿ Смотрит человек (опытный инженер отдела мониторинга)
- ⦿ Используем: Kibana, Grafana, Zabbix, руки и светлую голову 😊
- ⦿ Помимо пунктов из регламента, рассматриваются аномалии в графиках/мониторах/дашбордах
- ⦿ Всегда есть бизнес-проверки и бизнес-метрики



# А как проверять?

- ⊙ Индивидуально (сервисы очень разные)
- ⊙ Подробный письменный регламент проверки при деплое (обязательно)
- ⊙ Смотрит человек (опытный инженер отдела мониторинга)
- ⊙ Используем: Kibana, Grafana, Zabbix, руки и светлую голову 😊
- ⊙ Помимо пунктов из регламента, рассматриваются аномалии в графиках/мониторах/дашбордах
- ⊙ Всегда есть бизнес-проверки и бизнес-метрики
- ⊙ Проверяется каждый шаг деплоя

# А как проверять?

- ⊙ Индивидуально (сервисы очень разные)
- ⊙ Подробный письменный регламент проверки при деплое (обязательно)
- ⊙ Смотрит человек (опытный инженер отдела мониторинга)
- ⊙ Используем: Kibana, Grafana, Zabbix, руки и светлую голову 😊
- ⊙ Помимо пунктов из регламента, рассматриваются аномалии в графиках/мониторах/дашбордах
- ⊙ Всегда есть бизнес-проверки и бизнес-метрики
- ⊙ Проверяется каждый шаг деплоя
- ⊙ При малейших подозрениях на проблемы – максимально быстрый откат (в случае с blue/green – это секунды)

# Инструменты деплоя



\\ Богатыри \\ Виктор Васнецов

- ◎ Awx (Ansible) -> 49 job templates
- ◎ Jenkins (Groovy) -> 41 jobs
- ◎ Fabric (Python) -> 5 scripts

# Что еще автоматизировать?

\\ Бурлаки на Волге \\ Илья Репин





# Что еще автоматизировать?

\\ Бурлаки на Волге \\ Илья Репин



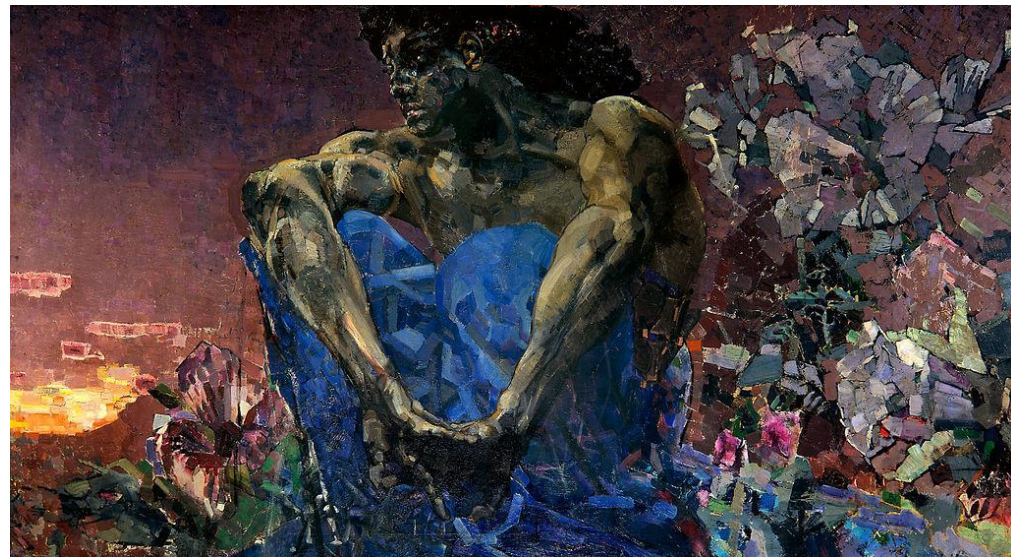
# Что еще автоматизировать?

\\ Бурлаки на Волге \\ Илья Репин



- ⊙ Работу деплой-инженера
- ⊙ Работу инженера мониторинга

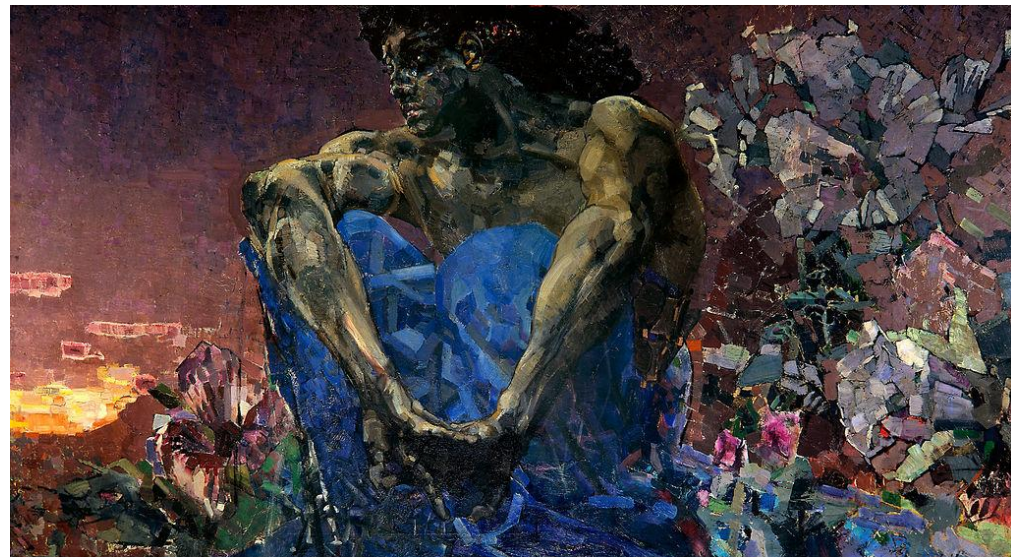
# Как автоматизировать?



\\ Демон сидящий \\ Михаил Врубель



# Как автоматизировать?



\\ Демон сидящий \\ Михаил Врубель

# Как автоматизировать?

 **TeamCity**

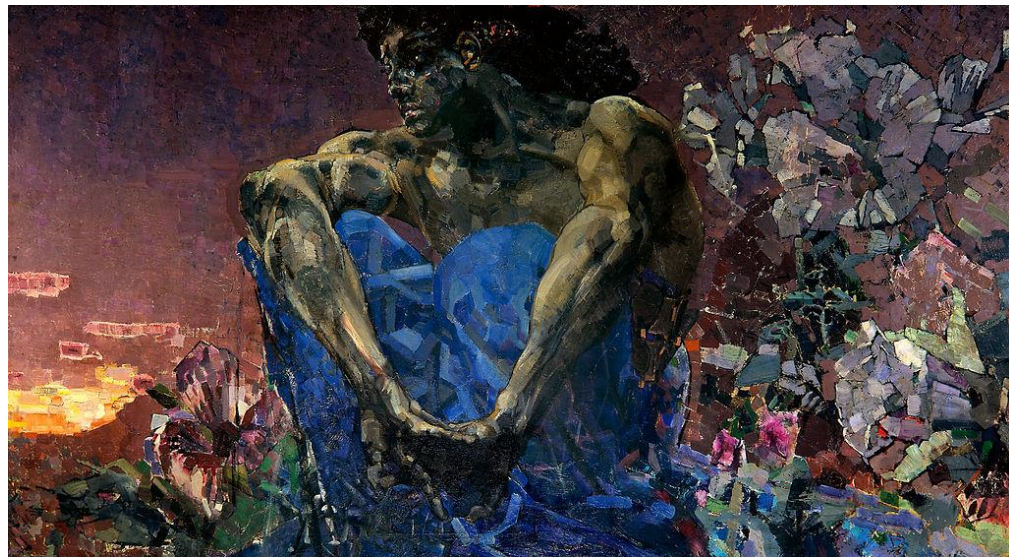


 **Bamboo**

 **StackStorm**

 **GitLab CI**

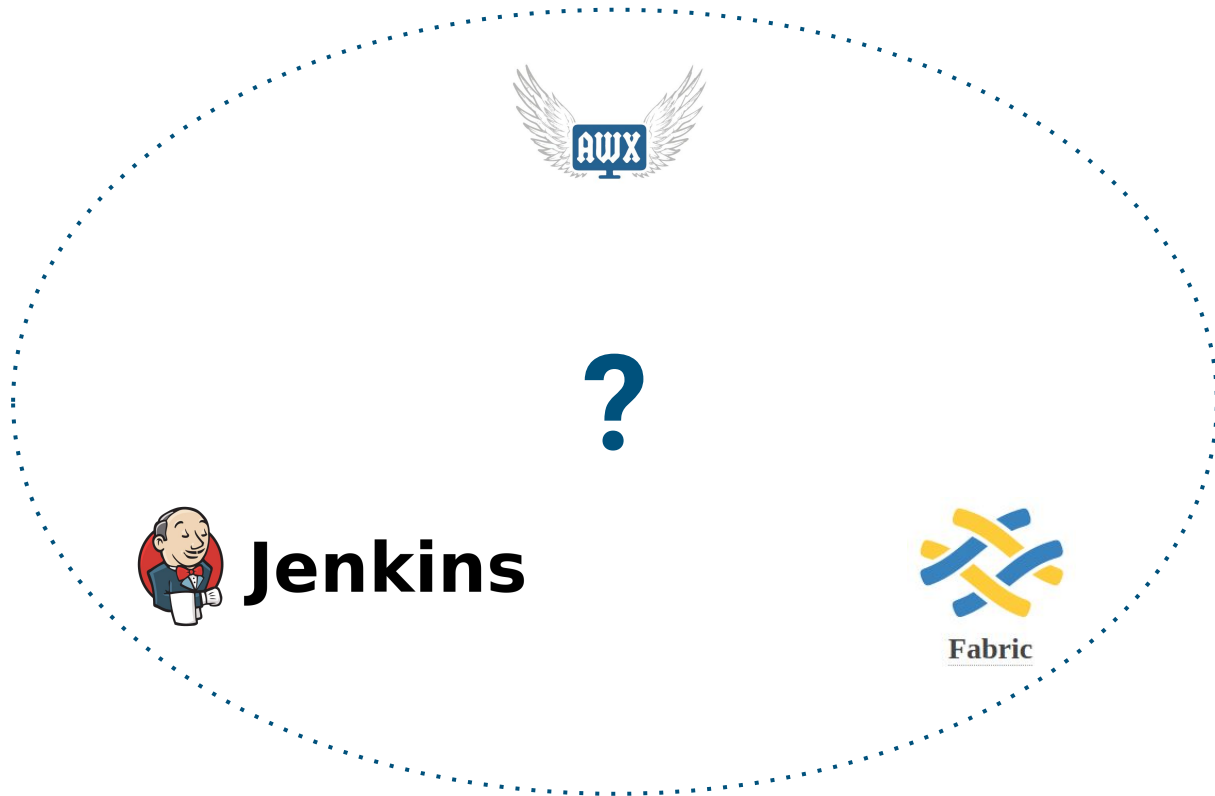
 **Jenkins**



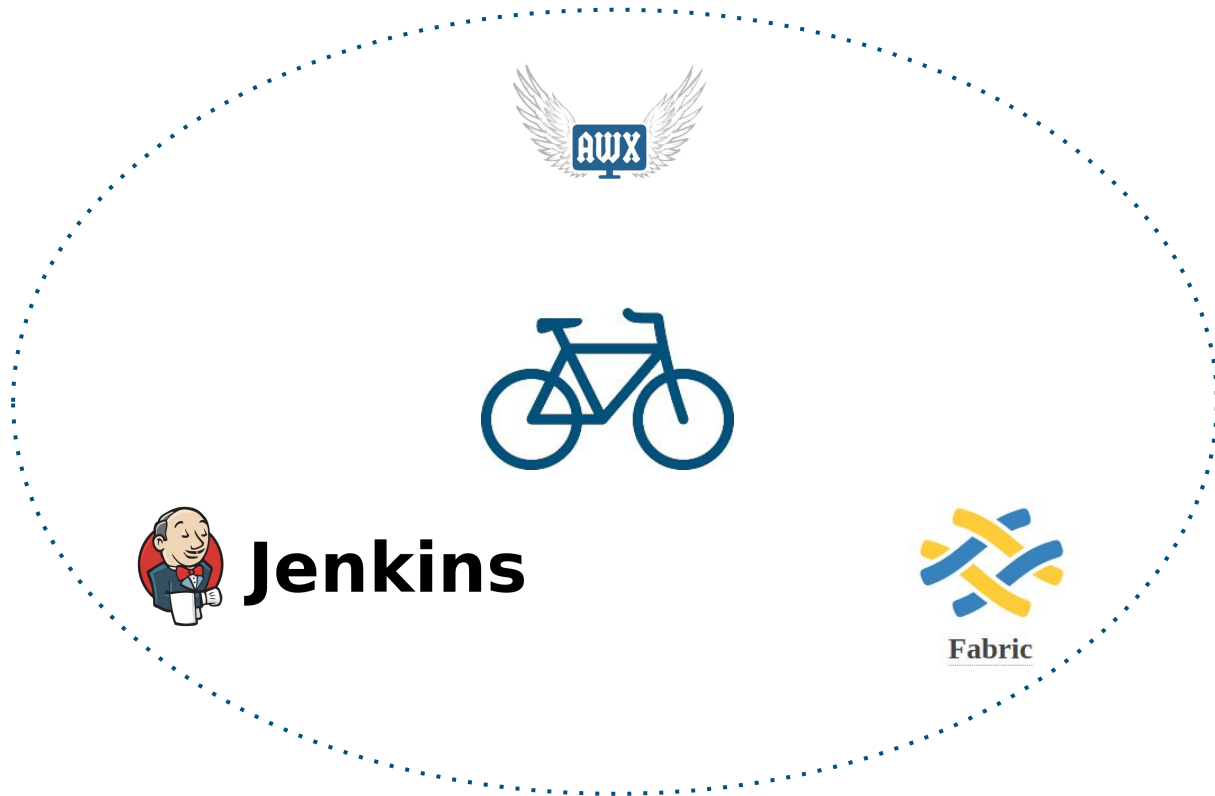
\\ Демон сидящий \\ Михаил Врубель



# Как автоматизировать?



# Как автоматизировать?



# Как автоматизировать?



# Как автоматизировать?



# Как автоматизировать?



© Blizzard Entertainment  
© Mr·Jack·09

# Как автоматизировать?



© Blizzard Entertainment  
© Mr·Jack·09

## Lurker

- © Соглядатай, тайный наблюдатель
- © Но если надо, то и пнуть может!



# Lurker

- ⊙ Отдельная система (оркестратор)

# Lurker

- ⊙ Отдельная система (оркестратор)
- ⊙ PHP 7.4, MySQL 5.7, Bootstrap

# Lurker

- ⊙ Отдельная система (оркестратор)
- ⊙ PHP 7.4, MySQL 5.7, Bootstrap
- ⊙ Jenkins API – для деплоев (включает fabric)

# Lurker

- ⊙ Отдельная система (оркестратор)
- ⊙ PHP 7.4, MySQL 5.7, Bootstrap
- ⊙ Jenkins API – для деплоев (включает fabric)
- ⊙ AWX API – для манипуляций с трафиком

# Lurker

- ⊙ Отдельная система (оркестратор)
- ⊙ PHP 7.4, MySQL 5.7, Bootstrap
- ⊙ Jenkins API – для деплоев (включает fabric)
- ⊙ AWX API – для манипуляций с трафиком
- ⊙ Мониторит состояние сервисов по логам в ELK (Elasticsearch API)

# Lurker

- ⊙ Отдельная система (оркестратор)
- ⊙ PHP 7.4, MySQL 5.7, Bootstrap
- ⊙ Jenkins API – для деплоев (включает fabric)
- ⊙ AWX API – для манипуляций с трафиком
- ⊙ Мониторит состояние сервисов по логам в ELK (Elasticsearch API)
- ⊙ Grafana API (считывает триггеры)



# Lurker

- ⊙ Отдельная система (оркестратор)
- ⊙ PHP 7.4, MySQL 5.7, Bootstrap
- ⊙ Jenkins API – для деплоев (включает fabric)
- ⊙ AWX API – для манипуляций с трафиком
- ⊙ Мониторит состояние сервисов по логам в ELK (Elasticsearch API)
- ⊙ Grafana API (считывает триггеры)
- ⊙ Zabbix API (считывает триггеры)

# Lurker

- ⊙ Отдельная система (оркестратор)
- ⊙ PHP 7.4, MySQL 5.7, Bootstrap
- ⊙ Jenkins API – для деплоев (включает fabric)
- ⊙ AWX API – для манипуляций с трафиком
- ⊙ Мониторит состояние сервисов по логам в ELK (Elasticsearch API)
- ⊙ Grafana API (считывает триггеры)
- ⊙ Zabbix API (считывает триггеры)
- ⊙ Запускает бизнесовые проверки сервисов (кастомные, PHP)


# Lurker

- ⊙ Отдельная система (оркестратор)
- ⊙ PHP 7.4, MySQL 5.7, Bootstrap
- ⊙ Jenkins API – для деплоев (включает fabric)
- ⊙ AWX API – для манипуляций с трафиком
- ⊙ Мониторит состояние сервисов по логам в ELK (Elasticsearch API)
- ⊙ Grafana API (считывает триггеры)
- ⊙ Zabbix API (считывает триггеры)
- ⊙ Запускает бизнесовые проверки сервисов (кастомные, PHP)
- ⊙ Запускает “Lurker-Smoke-тесты” (Codeception, Selenium WebDriver)

# Lurker

- ⊙ Отдельная система (оркестратор)
- ⊙ PHP 7.4, MySQL 5.7, Bootstrap
- ⊙ Jenkins API – для деплоев (включает fabric)
- ⊙ AWX API – для манипуляций с трафиком
- ⊙ Мониторит состояние сервисов по логам в ELK (Elasticsearch API)
- ⊙ Grafana API (считывает триггеры)
- ⊙ Zabbix API (считывает триггеры)
- ⊙ Запускает бизнесовые проверки сервисов (кастомные, PHP)
- ⊙ Запускает “Lurker-Smoke-тесты” (Codeception, Selenium WebDriver)
- ⊙ Web-интерфейс, Telegram-bot

# Lurker



Lurker

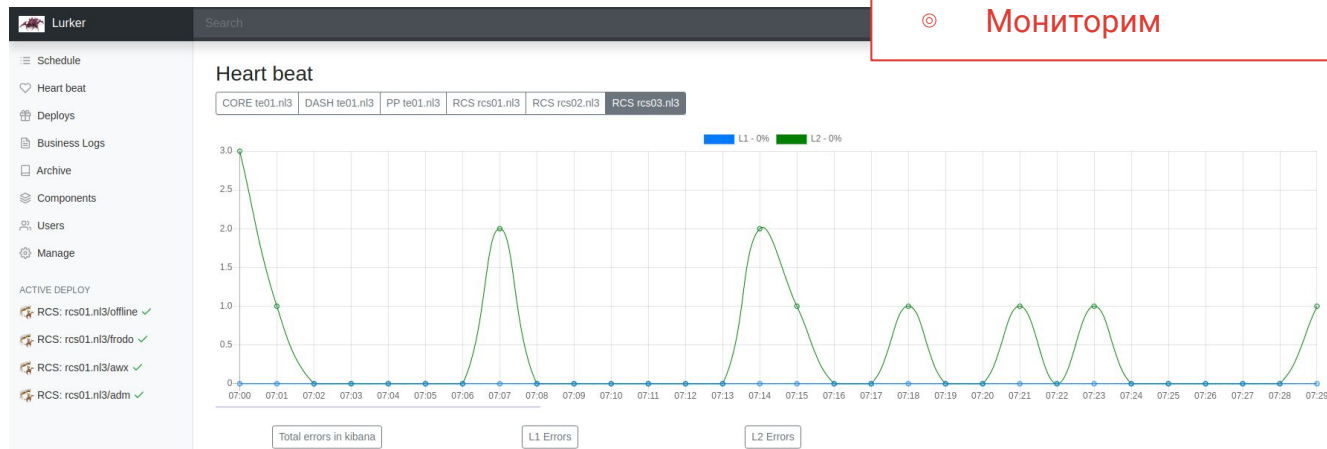
Fri, 7 May 2021 Date ▾

Time	MSK ▾	Deploy	Jira Release	Release Notes	Deploy Notes	Owner
10:00 - 10:30		<button>add deploy</button>				
10:30 - 11:00		<button>add deploy</button>				
11:00 - 11:30		<a href="#">#108 CORE</a> <span>🕒</span>	<a href="#">HOST/versions/22890</a>	1. Доработка кода мастервара для разрешения прохождения сценки после неудачного завершения. 2. Доработка кода мастервара для исключения из группы full сценки неудачный audit.	Загруженные системы и компоненты в деплои, включая на них - включая на автоматизированный график мастервара - автоматизированный Release и их автоматизации - получение логики проведения сценки (автоматизация для контроля прохождения сценки (сценария))	<a href="#">Fred</a>
11:30 - 12:00		<button>add deploy</button>				
12:00 - 12:30		<button>add deploy</button>				
12:30 - 13:00		<button>add deploy</button>				
13:00 - 13:30		<button>add deploy</button>				

- QA сами подают заявку на продовый деплой
- Расписание ведется прямо в Lurker
- SRE только присматривают
- Деплои запускаются автоматически
- Анализ и откат “в случае чего” – автоматически

# Lurker

Мониторим





# Lurker

**Business Logs**

☐ Auto refresh

**Job log**

```
[10:14:33 MSK] Job#301 RCS >rcs01.n13 Pre deploy check is successful (I)
[10:14:34 MSK] Job#301 RCS >rcs01.n13 Jenkins previous successful build is, Line: L1, Tag: lurker-5.92 (I)
[10:14:34 MSK] Job#301 RCS >rcs01.n13 No lines for job (I)
[10:14:39 MSK] Job#301 RCS >rcs01.n13 Job was runned (2021.04.29) (D)
[10:14:39 MSK] Job#301 RCS >rcs01.n13 Jenkins build was started, Job: /job/rcs01.n13/job/adm/, Tag: rc-5.93, Line: L1 (D)
[10:17:14 MSK] Job#301 RCS >rcs01.n13 (1) All traffic on new code. Check start immediately (D)
[10:17:14 MSK] Job#301 RCS >rcs01.n13 (1) Checking... (D)
[10:17:15 MSK] Job#301 RCS >rcs01.n13 (1) Alright. RcsChecks->errorsFull: is OK (D)
[10:17:15 MSK] Job#301 RCS >rcs01.n13 (1) Waiting. RcsChecks->frontTest: waiting until 600 seconds (D)
[10:17:15 MSK] Job#301 RCS >rcs01.n13 (1) Skip. RcsChecks->grafanaTriggers: without check (D)
[10:17:25 MSK] Job#301 RCS >rcs01.n13 (1) Alright. RcsChecks->errorsFull: is OK (D)
[10:17:25 MSK] Job#301 RCS >rcs01.n13 (1) Waiting. RcsChecks->frontTest: waiting until 600 seconds (D)
[10:17:25 MSK] Job#301 RCS >rcs01.n13 (1) Skip. RcsChecks->grafanaTriggers: without check (D)
[10:17:35 MSK] Job#301 RCS >rcs01.n13 (1) Alright. RcsChecks->errorsFull: is OK (D)
[10:17:35 MSK] Job#301 RCS >rcs01.n13 (1) Waiting. RcsChecks->frontTest: waiting until 600 seconds (D)
[10:17:35 MSK] Job#301 RCS >rcs01.n13 (1) Skip. RcsChecks->grafanaTriggers: without check (D)
[10:17:45 MSK] Job#301 RCS >rcs01.n13 (1) Alright. RcsChecks->errorsFull: is OK (D)
[10:17:45 MSK] Job#301 RCS >rcs01.n13 (1) Waiting. RcsChecks->frontTest: waiting until 600 seconds (D)
[10:17:46 MSK] Job#301 RCS >rcs01.n13 (1) Skip. RcsChecks->grafanaTriggers: without check (D)
[10:17:56 MSK] Job#301 RCS >rcs01.n13 (1) Alright. RcsChecks->errorsFull: is OK (D)
[10:17:56 MSK] Job#301 RCS >rcs01.n13 (1) Waiting. RcsChecks->frontTest: waiting until 600 seconds (D)
[10:17:56 MSK] Job#301 RCS >rcs01.n13 (1) Skip. RcsChecks->grafanaTriggers: without check (D)
[10:18:06 MSK] Job#301 RCS >rcs01.n13 (1) Alright. RcsChecks->errorsFull: is OK (D)
[10:18:06 MSK] Job#301 RCS >rcs01.n13 (1) Waiting. RcsChecks->frontTest: waiting until 600 seconds (D)
[10:18:06 MSK] Job#301 RCS >rcs01.n13 (1) Skip. RcsChecks->grafanaTriggers: without check (D)
[10:18:16 MSK] Job#301 RCS >rcs01.n13 (1) Alright. RcsChecks->errorsFull: is OK (D)
[10:18:16 MSK] Job#301 RCS >rcs01.n13 (1) Alright. RcsChecks->frontTest: is OK (D)
[10:18:16 MSK] Job#301 RCS >rcs01.n13 (1) Skip. RcsChecks->grafanaTriggers: without check (D)
[10:18:16 MSK] Job#301 RCS >rcs01.n13 (1) End of check step: Alright (D)
[10:18:16 MSK] Job#301 RCS >rcs01.n13 (1) Job was successful (D)
[10:18:16 MSK] Global Try to run next scheduled job
[10:18:16 MSK] Global No jobs in active deploy for run
```

Мониторим

Смотрим за деплоем



# Lurker

The screenshot shows the Lurker application interface. On the left is a sidebar with navigation options: Schedule, Heart beat, Deployments, Business Logs, Archive, Components, Users, Manage, and ACTIVE DEPLOY. The main panel displays 'Deploy #97 RCS: Thu, 29 Apr 2021, 14:00 MSK'. Below this is a table with columns 'Job type', 'Params', and 'Actions'. The table lists four jobs: 'RCS: rcs01.nl3/offline', 'RCS: rcs01.nl3/frodo', 'RCS: rcs01.nl3/awx', and 'RCS: rcs01.nl3/adm'. Below the table is an 'Add job' section with fields for Component (RCS), Job template (rcs01.nl3/adm), Tag (Tag), and Line (L1), and an 'Add' button. At the bottom is an 'Add manual action' button.

Job type	Params	Actions
RCS: rcs01.nl3/offline ✓	tag: rc-5.18, line: L1	[Play] [Pause] [Stop] [Up] [Down] [Trash]
RCS: rcs01.nl3/frodo ✓	tag: rc-5.76, line: auto	[Play] [Pause] [Stop] [Up] [Down] [Trash]
RCS: rcs01.nl3/awx ✓	line: auto	[Play] [Pause] [Stop] [Up] [Down] [Trash]
RCS: rcs01.nl3/adm ✓	tag: rc-5.93, line: L1	[Play] [Pause] [Stop] [Up] [Down] [Trash]

**Add job**

Component: RCS

Job template: rcs01.nl3/adm

Tag: Tag

Line: L1

**Add**

**Add manual action**

- Мониторим
- Смотрим за деплоем
- Управляем...

# Что еще важно?



\\ Витязь на распутье \\ Виктор Васнецов

# Что еще важно?

- © Zero Tolerance к ошибкам уровней ERROR+ (перманентный процесс) – очень помогает мониторить деплои



\\ Витязь на распутье \\ Виктор Васнецов

# Что еще важно?

- © Zero Tolerance к ошибкам уровней ERROR+ (перманентный процесс) – очень помогает мониторить деплои
- © Тестирование совместимости разных версий сервисов силами QA



\\ Витязь на распутье \\ Виктор Васнецов



# Что еще важно?

- ◎ Zero Tolerance к ошибкам уровней ERROR+ (перманентный процесс) – очень помогает мониторить деплои
- ◎ Тестирование совместимости разных версий сервисов силами QA
- ◎ Линий для DBMS – нет



\\ Витязь на распутье \\ Виктор Васнецов



# Что еще важно?

- ◎ Zero Tolerance к ошибкам уровней ERROR+ (перманентный процесс) – очень помогает мониторить деплои
- ◎ Тестирование совместимости разных версий сервисов силами QA
- ◎ Линий для DBMS – нет
- ◎ Базы не откатываются



\\ Витязь на распутье \\ Виктор Васнецов

# Что еще важно?

- ⊙ Zero Tolerance к ошибкам уровней ERROR+ (перманентный процесс) – очень помогает мониторить деплои
- ⊙ Тестирование совместимости разных версий сервисов силами QA
- ⊙ Линий для DBMS – нет
- ⊙ Базы не откатываются
- ⊙ Обратная совместимость гарантируется приложением, проверяется QA



\\ Витязь на распутье \\ Виктор Васнецов

# Что еще важно?

- ⊙ Zero Tolerance к ошибкам уровней ERROR+ (перманентный процесс) – очень помогает мониторить деплои
- ⊙ Тестирование совместимости разных версий сервисов силами QA
- ⊙ Линий для DBMS – нет
- ⊙ Базы не откатываются
- ⊙ Обратная совместимость гарантируется приложением, проверяется QA
- ⊙ Не боимся



\\ Витязь на распутье \\ Виктор Васнецов





Devops'ы на конференции:

- как сделать, чтобы релизить быстро и качественно?
- могу рассказать
- рассказать я и сам могу, а как сделать то?



Фёдор

в сети

телега: @fred\_msk

f.vasiliev@it.ecommpay.com

fedor.vasiliev@gmail.com

Спасибо за Внимание!

Devops'ы на конференции:

- как сделать, чтобы релизить быстро и качественно?
- могу рассказать
- рассказать я и сам могу, а как сделать то?